## LISTING OF THE CLAIMS:

1. (Currently Amended) A method in a data processing system for transferring data, the method comprising:

recciving data describing objects in an original format, wherein the original format is unusable by a set of processing environments, and wherein the data comprises a graphical user interface object hierarchy comprising a sequence of functional units for execution in a graphical user interface;

storing the data in a database format in a database;

responsive to receiving a requcst for an object from a particular processing environment within thc set of processing environments, retrieving data corresponding to the objcct from the database; and

translating the data corresponding to the object into a selected format ~~form~~ usable by thc particular ~~for use by the~~ processing environment.

2. (Currently Amended) The method of claim 1, ~~wherein the data is in a markup language data file~~ wherein the object contains a functional unit presented as an action to a user in the particular processing environment via a graphical user interface console, wherein the action has a same look and feel for each of the set of processing environments.

3. (Currently Amended) The method of claim 1, wherein the database format is a set of entries in a tablc, and whercin the database format corresponds to information obtaincd from the graphical user interface hierarchy.

4. (Currently Amended) The method of claim 2, ~~wherein the request originates from a data proxy~~ wherein the data describes a relationship between an instance of an action object and an instance of a functional unit definition object.

5. (Original) The method of claim 1, wherein the proccssing environment is a Java processing environment and the form is a Java class.

6. (Currently Amended) A method in a data proccssing system for transferring data, the method comprising:

rcceiving a markup language file describing at least onc object in an original format, wherein the original format is unusable by a sct of requestors, and wherein the data comprises a graphical user interface objcct hierarchy comprising a sequence of functional units for cxecution in a graphical user interface;

converting the markup language filc to at least one table in a database, wherein the at least one table contains object parameters for the at least onc object;

rcsponsive to a request for an object from a particular requestor within the set of requestors, translating the at least one table into the object into a selected format usable by the particular requestor; and

sending the object to the particular requestor.

7. (Original) The method of claim 6, further comprising:

validating the markup language file using a document type definition file prior to the prior to converting the markup language file.

8. (Currently Amended) The method of claim 6, wherein the request for the object is for an object in a desired form selected among a plurality of available object forms and wherein the ~~tranlating~~ translating step translates the table into the desired object form.

9. (Original) Thc method of claim 6, wherein the object is a graphical user interface object used for representing a system resource in a graphical user interface.

10. (Currcntly Amcndcd) A method in a data processing system for transferring data, the method comprising:

storing external customizable data for use by a set of software ~~system~~ systems during execution of processes by the softwarc system in a central repository, wherein the set of software ~~system~~ systems are [[is]] distributed within a network data processing system; and delivering the external customizable data in a selected format usable by a particular software system in the set of software systems in response to requests from the particular software system.

11. (Original) The method of claim 10, wherein the external customizable data is an extensible markup language data file.

12. (Original) The method of claim 10, wherein the object is an instance of a Java class.

13. (Original) The method of claim 10, wherein the requestor is a data proxy.

14. (Original) The method of claim 10, wherein the object is a Java class.

15. (Original) The method of claim 10, wherein the object is an instance of a Java object.

16. (Original) The method of claim 10, wherein the step of sending the object to the requestor comprises:

sending a universal resource identifier to the requestor.

17. (Original) The method of claim 10, wherein the external customizable data is a markup language file and further comprising:

validating the markup language file.

18. (Original) The method of claim 17, wherein the markup language file is an extensible markup language file.

19. (Original) The method of claim 18, wherein the extensible markup language file is validated using a document type definition file.

20. (Currently Amended) A system for transferring data, the system comprising:

a database, wherein the database contains representations of objects;

a data import process, wherein the data import process receives an external data file describing an object in an original format, wherein the original format is unusable by a set of requestors, and wherein the external data file comprises a graphical user interface object hierarchy comprising a sequence of functional units for execution in a graphical user interface;, translates external data file into a representation, and stores the representation in the database; and

a data server process, wherein the data server process receives a request from a particular requestor in the set of requestors, fetches a selected representation in response to receiving the request, translates the selected representation into an object into a selected format usable by the particular requestor, and sends the object to the particular requestor.

21. (Original) The system of claim 20 further comprising:

a set of data proxies, wherein a data proxy within the set of data proxies connects to the data server process, receives a request from a local processing environment, routes the request to the data server, receives a result from the data server process, and sends the result to the local processing environment.

22. (Original) The system of claim 20, wherein the external data file is a markup language file.

23. (Original) The system of claim 20, wherein the markup language file is an extensible markup language file.

24. (Currently Amended) A data processing system comprising:

a bus system;

a communications unit connected to the bus, wherein data is sent and received using the communications unit;

a memory connected to the bus system, wherein a set of instructions is located in the memory; and

a processor unit connected to the bus system, wherein the processor unit executes the set of instructions to receive data describing objects in an original format, wherein the original format is unusable by a set of processing environments, and wherein the data comprises a graphical user interface object hierarchy comprising a sequence of functional units for execution in a graphical user interface; store the data in a database format in a database; retrieve data corresponding to the object from the database in response to receiving a request for an object from a particular processing environment within the set of processing environments; and translate the data corresponding to the object into a ~~form~~ selected format usable by the particular ~~for use by the~~ processing environment.

25. (Original) The data processing system of claim 24, wherein the bus system includes a primary bus and a secondary bus.

26. (Original) The data processing system of claim 24, wherein the processor unit includes a single processor.

27. (Original) The data processing system of claim 24, wherein the processor unit includes a plurality of processors.

28. (Original) The data processing system claim 24, wherein the communications unit is an Ethernet adapter.

29. (Currently Amended) A data processing system comprising:

a bus system;

a communications unit connected to the bus, wherein data is sent and received using the communications unit;

a memory connected to the bus system, wherein a set of instructions is located in the memory; and

a processor unit conncctcd to the bus system, wherein the processor unit cxecutes the set of instructions to receive a markup language file describing at least one object in an original format, wherein the original format is unusable by a set of requestors, and wherein the markup language file comprises a graphical user interface object hierarchy comprising a sequence of functional units for execution in a graphical user interface; convert the markup language file to at least one table in a database, wherein the at least onc tablc contains object parameters for the at least one object; translating the at least one table into the object in response to a request for an object from a particular requestor, wherein the at least one table is translated into the object in a format usable by the particular requestor in the set of requestors; and send the object to the particular requestor.

30. (Original) The data processing system of claim 29, wherein the bus system includes a primary bus and a secondary bus.

31. (Original) The data processing system of claim 29, wherein the processor unit includes a single processor.

32. (Original) The data processing system of claim 29, whercin thc processor unit includes a plurality of processors.

33. (Original) The data processing system claim 29, wherein the communications unit is an Ethernet adapter.

34. (Currently Amended) A data processing system comprising:

a bus system;

a communications unit connected to the bus, wherein data is sent and received using the communications unit;

a memory connected to the bus system, wherein a set of instructions is located in the memory; and

a processor unit connected to the bus system, wherein the processor unit executes the set of instructions to store external customizable data for use by a set of software ~~system~~ systems during execution of processes by the set of software ~~system~~ systems in a central repository, wherein the set of software ~~system~~ systems are [[is]] distributed within a network data processing system; and deliver the external customizable data in a selected format usable by [[the]] a particular software system in the set of software systems in response to requests from the particular software system.

35. (Original) The data processing system of claim 34, wherein the bus system includes a primary bus and a secondary bus.

36. (Original) The data processing system of claim 34, wherein the processor unit includes a single processor.

37. (Original) The data processing system of claim 34, wherein the processor unit includes a plurality of processors.

38. (Original) The data processing system claim 34, wherein the communications unit is an Ethernet adapter.

39. (Currently Amended) A data processing system for transferring data, the data processing system comprising:

receiving means for receiving data describing objects in an original format, wherein the original format is unusable by a set of processing environments, and wherein the data comprises a graphical user interface object hierarchy comprising a sequence of functional units for execution in a graphical user interface;

storing means for storing the data in a database format in a database;

retrieving means, responsive to receiving a request for an object from a particular processing environment within the set of processing environments, for retrieving data corresponding to the object from the database; and

translating means for translating the data corresponding to the object into a selected format ~~form~~ usable by the particular ~~for use by the~~ processing environment.

40. (Original) The data processing system of claim 39, wherein the data is in a markup language data file.

41. (Original) The data processing system of claim 39, wherein the database format is a set of entries in a table.

42. (Original) The data processing system of claim 39, wherein the request originates from a data proxy.

43. (Currently Amended) The data processing ~~sysytem~~ system of claim 39, wherein the processing environment is a Java processing environment and the form is a Java class.

44. (Currently Amended) A data processing system for transferring data, the data processing system comprising:

receiving means for receiving a markup language file describing at least one object in an original format, wherein the original format is unusable by a set of requestors, and wherein the markup language file comprises a graphical user interface object hierarchy comprising a sequence of functional units for execution in a graphical user interface;

converting means for converting the markup language file to at least one table in a database, wherein the at least one table contains object parameters for the at least one object;

translating means, responsive to a request for an object from a particular requestor in the set of requestors, for translating the at least one table into the object into a selected format usable by the particular requestor; and

sending means for sending the object to the particular requestor.

45. (Original) The data processing system of claim 44, further comprising:

validating means for validating the markup language file using a document type definition file prior to the prior to converting the markup language file.

46. (Currently Amended) The data processing system of claim 44, wherein the request for the object is for an object in a desired form selected among a plurality of available object forms and wherein the ~~tranlating~~ translating step translates the table into the desired object form.

47. (Original) The data processing system of claim 44, wherein the object is a graphical user interface object used for representing a system resource in a graphical user interface.

48. (Currently Amended) A data processing system for transferring data, the data processing system comprising:

storing means for storing external customizable data for use by a set of software ~~system~~ systems during execution of processes by the set of software ~~system~~ systems in a central repository, wherein the set of software systems are ~~system is~~ distributed within a network data processing system; and

delivering means for delivering the external customizable data in a selected format usable by [[the]] a particular software system in the set of software systems in response to requests from the particular software system.

49. (Original) The data processing system of claim 48, wherein the external customizable data is an extensible markup language data file.

50. (Original) The data processing system of claim 48, wherein the object is an instance of a Java class.

51. (Original) The data processing system of claim 48, wherein the requestor is a data proxy.

52. (Original) The data processing system of claim 48, wherein the object is a Java class.

53. (Original) The data processing system of claim 48, wherein the sending means comprises:

means for sending a universal resource identifier to the requestor.

54. (Original) The data processing system of claim 48, wherein the external customizable data is a markup language file and further comprising:

validating means for validating the markup language file.

55. (Original) The data processing system of claim 54, wherein the markup language file is an extensible markup language file.

56. (Original) The data processing system of claim 55, wherein the extensible markup language file is validated using a document type definition file.

57. (Currently Amended) A computer program product in a computer readable medium for transferring data, the method comprising:

first instructions for receiving data describing objects in an original format, wherein the original format is unusable by a set of processing environments, and wherein the data comprises a graphical user interface object hierarchy comprising a sequence of functional units for execution in a graphical user interface;

second instructions for storing the data in a database format in a database;

third instructions, responsive to receiving a request for an object from a particular processing environment within the set of processing environments, for retrieving data corresponding to the object from the database; and

fourth instructions for translating the data corresponding to the object into a selected format ~~form~~ usable by the particular ~~for use by the~~ processing environment.

58. (Currently Amended) A computer program product in a computer readable medium for transferring data, the computer program product comprising:

first instructions for receiving a markup language file describing at least one object in a original format, wherein the original format is unusable by a set of requestors, and wherein the markup language file comprises a graphical user interface object hierarchy comprising a sequence of functional units for execution in a graphical user interface;

second instructions for converting the markup language file to at least one table in a database, wherein the at least one table contains object parameters for the at least one object;

third instructions, responsive to a request for an object from a particular requestor in the set of requestors, for translating the at least one table into the object into a selected format usable by the particular requestor; and

fourth instructions for sending the object to the particular requestor.

59. (Currently Amended) A computer program product in a computer readable medium for use in transferring data, the computer program product comprising:

first instructions for storing external customizable data for use by a set of software ~~system~~ systems during execution of processes by the set of software ~~system~~ systems in a central repository, wherein the set of software systems are ~~system is~~ distributed within a network data processing system; and

second instructions for delivering the external customizable data in a selected format usable by [[the]] a particular software system in the set of software systems in response to requests from the particular software system.